AF IFW
2171

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Daniel D. McNeil

Assignee: Mirapoint, Inc.

Title: METHOD AND SYSTEM FOR PROVIDING IMAGE INCREMENTAL AND DISASTER RECOVERY

Serial No.: 10/066,109 File Date: January 31, 2002

Examiner: Patrick JD Santos Art Unit: 2171

Docket No.: MPT-006

Date: March 28, 2005

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief, filed in triplicate, is in support of the Notice of Appeal dated March 15, 2005.

/
/
/
/
/
/
/
/
/
/
/

INDEX

I. REAL PARTY IN INTEREST

The real party in interest is the assignee, Mirapoint, Inc., pursuant to the Assignment recorded in the U.S. Patent and Trademark Office on January 31, 2002 on Reel 012576, Frame 0557.

II. RELATED APPEALS AND INTERFERENCES

Based on information and belief, there are no other appeals or interferences that could directly affect or be directly affected by or have a bearing on the decision by the Board of Patent Appeals in the pending appeal.

III. STATUS OF CLAIMS

Claims 1-24 are pending. Claims 1-24 stand rejected.

In the present paper, rejected Claims 1-24 are appealed.

Pending Claims 1-24 are listed in Appendix A.

IV. STATUS OF AMENDMENTS

Claims 1, 9, 13, and 21 were amended during the prosecution of this application. All claim amendments are now entered.

V. SUMMARY OF THE INVENTION

Figures 1A and 1B are shown below to facilitate understanding of Applicant's invention. As taught by Applicant in the Specification, paragraphs [0007] - [0011]:

> [0007] A method for backing up data in a computer system from at least one primary data source to a secondary data source is provided. The method includes performing a full image backup on a plurality of data blocks stored by the primary data source(s). An incremental backup can then be initiated at a predetermined interval. During this incremental backup, the modification time of each file and folder is examined. If the modification time is earlier than the defined time, then the data block used by that file/folder is added to an unused data block

list. All files/folders are examined in a similar manner. All blocks, except those data blocks in the unused list, can then be written to tape with their file system metadata.

[0008] Another method to accomplish this image incremental backup, is to examine the modification time of each file and folder, and list all data blocks associated with the files/folders whose modification time is later than the defined time in the incremental backup. All files/folders are examined in a similar manner. All blocks on the used list can then be written to tape with their file system metadata.

[0009] In either approach, this method creates an image incremental backup that includes the file system metadata and all data from files/folders that have changed since the last backup. The data is written in disk order and, because it does not contain data from files/folders that have not changed, the amount of data and the time it takes to write the data to tape is much smaller than a full image backup.

[0010] In one embodiment, the defined time is a time when the full image backup was performed. In another embodiment, the defined time is a time when a last incremental backup was performed. In yet another embodiment, the defined time is either a first time when the full image backup was performed or a second time when a last incremental backup was performed, whichever is the most recent.

[0011] Because file systems, by design, already track each file/folder's modification time, this metadata is available and can be tracked without any additional overhead during normal operation. Checking modification times only during the incremental backup eliminates the significant overhead associated with tracking blocks that change during normal operation.

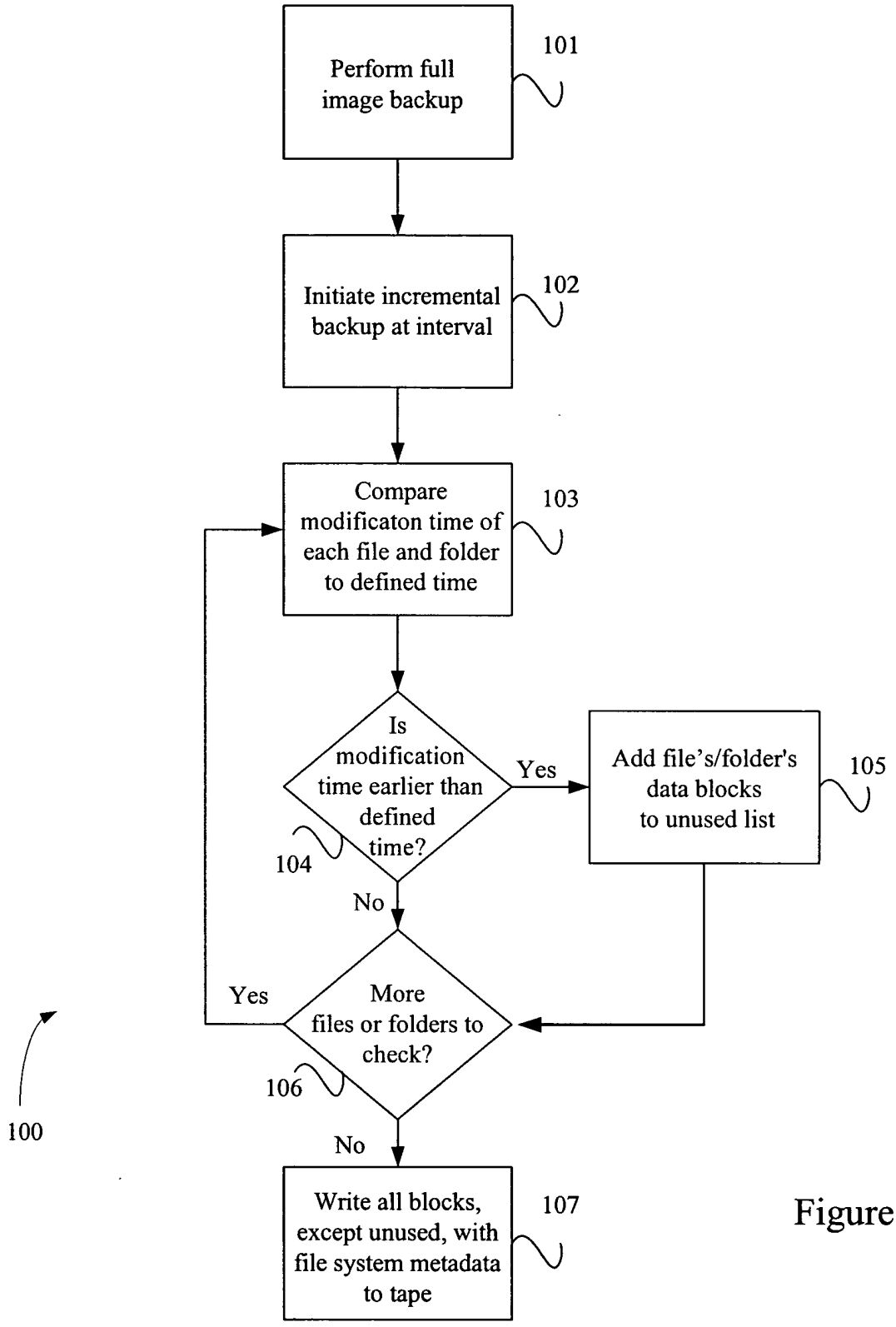
Perform full image backup
101
Initiate incremental backup at interval
102
Compare modificaton time of each file and folder to defined time
103
Is modification time earlier than defined time?
104
Yes
Add file's/folder's data blocks to unused list
105
No
More files or folders to check?
106
Yes
No
Write all blocks, except unused, with file system metadata to tape
107
100

Figure 1A

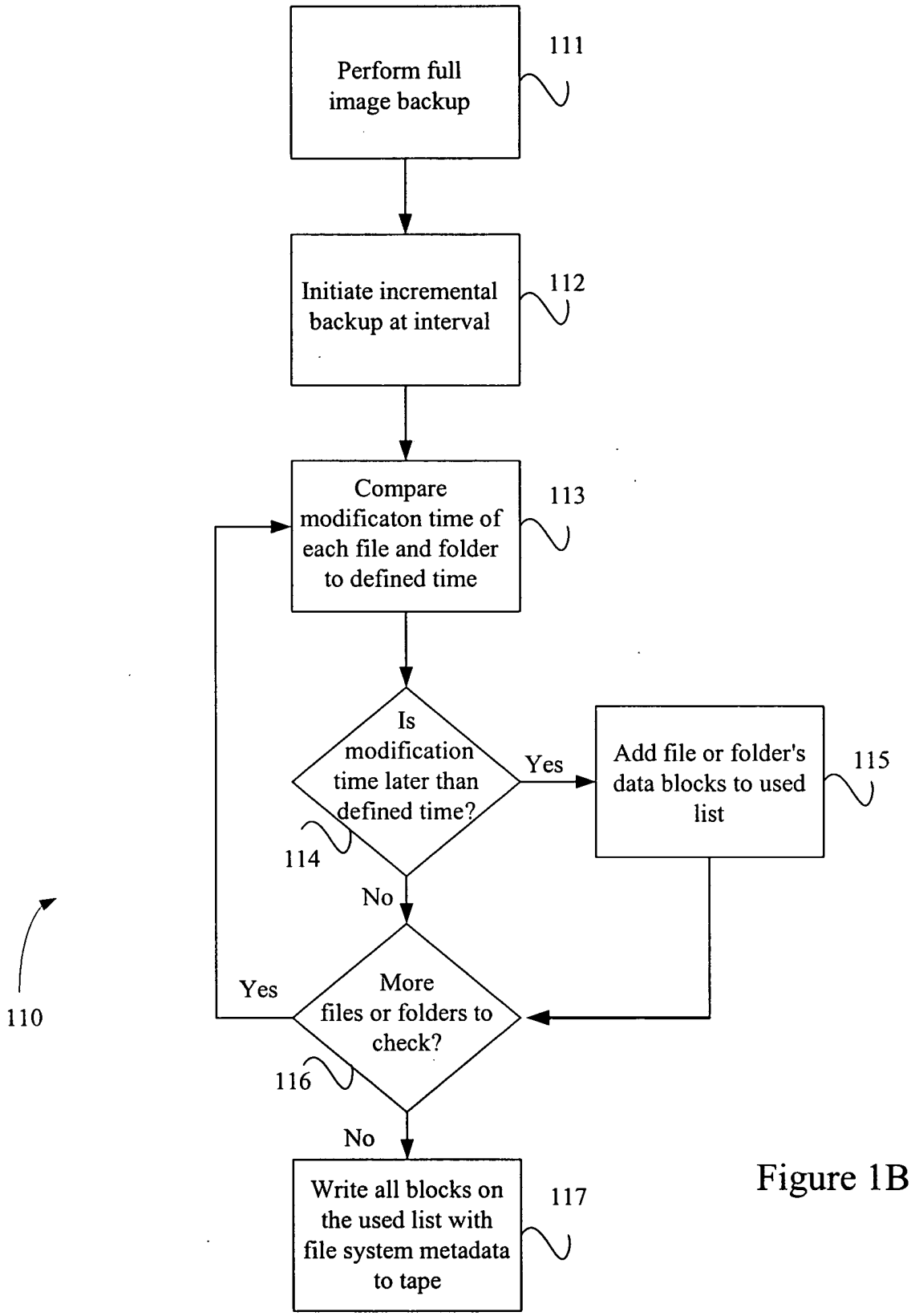
Perform full image backup
111
Initiate incremental backup at interval
112
Compare modificaton time of each file and folder to defined time
113
Is modification time later than defined time?
Yes
Add file or folder's data blocks to used list
115
114
No
More files or folders to check?
Yes
116
No
Write all blocks on the used list with file system metadata to tape
117
110

Figure 1B

As further taught by Applicant in the Specification, paragraphs [0022] - [0024]:

> [0022] Advantageously, because both of these image incremental backups include the file system metadata as well as the files and folders that have changed, all file system changes can be reflected in the backup. Specifically, all files and folders that are new, changed, removed, renamed, and linked are reflected in the image incremental backup.
>
> [0023] Therefore, of importance, including file system metadata in the backup significantly increases the accuracy of the backup compared to a standard file-by-file backup, which only identifies new/changed files. Moreover, because an image backup writes data in disk order, not file order, this backup is faster than a standard file-by-file backup. Finally, because each file's/folder's modification time is already part of the file system metadata being tracked and updated by the file system, this backup method has no associated overhead during normal operation.
>
> [0024] Advantageously, because an image incremental backup includes all file system metadata, this image incremental backup along with the last full image backup can be used to restore a system to the point in time of the last backup in the event of a disaster. Thus, image incremental backups along with the last full image provide an effective and efficient disaster recovery mechanism.

VI. ISSUES

The following issues are presented to the Board of Appeals for decision:

(A) Whether Claims 1, 9-13, and 21-24 are patentable under 35 U.S.C. 103(a) over U.S. Patent 5,799,147 (Shannon)

in view of U.S. Patent 5604862 (Midgely) and further in view of U.S. Patent 6,415,300 (Liu).

(B) Whether Claims 2-4 and 14-16 are patentable under 35 U.S.C. 103(a) over Shannon, Midgely, and Liu in view of the publication entitled "Oracle 7 Server Administrator's Guide" (Oracle).

(C) Whether Claims 5 and 17 are patentable under 35 U.S.C. 103(a) as being unpatentable over Shannon, Midegly, and Liu in view of U.S. Patent 5,195,025 (Boecker).

(D) Whether Claims 6-8 and 18-20 are patentable under 35 U.S.C. 103(a) as being unpatentable over Shannon, Midegly, Liu, and Boecker in view of Oracle.

VII. GROUPING OF THE CLAIMS

Claims 1-24 stand or fall together.

VIII. ARGUMENTS

(A) Claims 1, 9-13, and 21-24 are patentable under 35 U.S.C. 103(a) over U.S. Patent 5,799,147 (Shannon) in view of U.S. Patent 5604862 (Midgely) and further in view of U.S. Patent 6,415,300 (Liu).

1. Shannon Overview

As generally taught by Shannon in the Abstract:

> The invention relates to a computer file backup method, which method comprises providing at least one client computer, such as a personal computer, having a data storage means, such as a hard disk, with data stored thereon, on which data backup protection is desired, and providing at least one

separate server computer having a data storage means, such as a hard disk, as a backup computer to receive data from the client computer. The method provides for backing up and periodically updating information on personal computers with the server computer located in a remote geographical location, the computers being connected by a network system.

A portion of Fig. 2 of Shannon is shown below to facilitate further understanding of this reference.

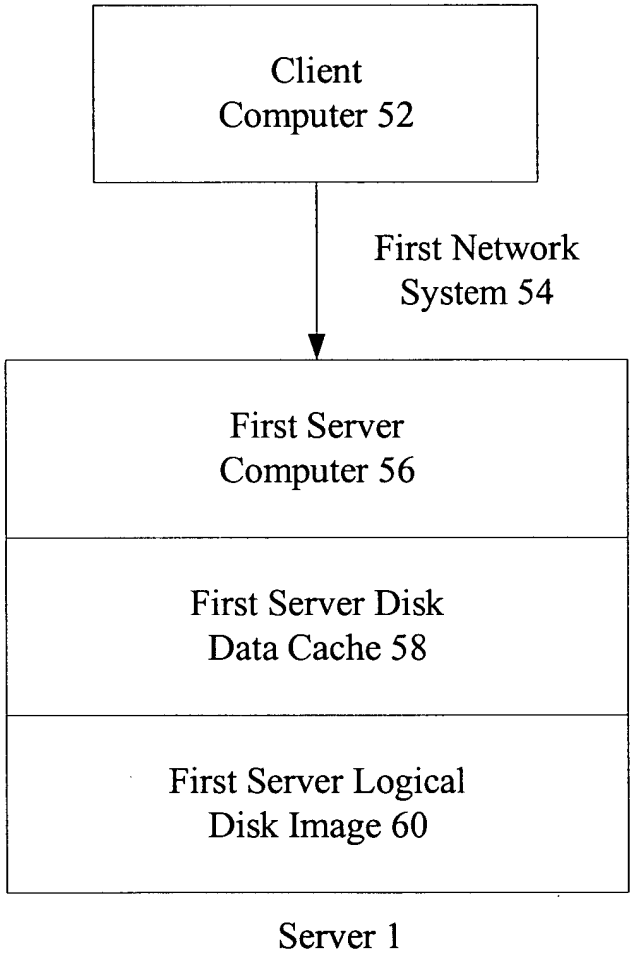

Fig. 2

As taught by Shannon in col. 3, lines 6-36

The method provides for creating a logical disk map of the client computer disk, and connecting the client computer to the server computer. The method further comprises copying a client computer logical disk image, including the logical disk map from the client computer disk, to the disk of the server

computer. The method provides for initiating by the client computer operator the updating of the disk map with the new disk map of the client computer, either manually by the computer operator or by a programmed, preselected automatic means, such as a preprogrammed code word or key sequence.

The connection between the client computer and the server computer having been severed, periodic updating of the disk map of the client computer by the computer user takes place, creating a new disk map, with the client computer comparing the disk map with the new disk map to create a list of modified files and removed files, which are themselves included in the list of modified files.

The method the provides for reconnecting the client computer to the server computer, and transmitting, generally by a publicly switched telecommunications network system, the modified files only from the disk of the client computer to a disk data cache on the server computer disk over the connection. Transferring of the data files from the disk data cache on the server disk to the server logical disk image is initiated, and the files identified as removed from the client disk are removed from the server logical disk image.

After completing the backup transmission, the client computer is notified of the update completion and the transmission connection between the client computer and the server computer is terminated.

2. Midgely Overview

As generally taught by Midgely in the Abstract:

An Integrity Server computer for economically protecting the data of a computer network's servers, and providing hot standby access to up-to-date copies of the data of a failed server. As the servers' files are created or modified, they are copied to the Integrity Server. The invention provides novel methods for managing the data stored

on the Integrity Server, so that up-to-date snapshots of files of the protected file servers are stored on low-cost media such as tape, but without requiring that a system manager load large numbers of tapes.

Fig. 1 of Midgely is shown below to facilitate further understanding of this reference. As taught by Midgely in col. 3, line 60 to col. 4, line 57:

> When all file servers 102 under the protection of Integrity Server 100 are operational (FIGS. 1 and 2a), the system operates in protection mode: Integrity Server 100 receives up-to-date copies of the protected files of the servers 102. When any protected server 102 goes down (FIGS. 1 and 2b), the system operates in stand-in mode: Integrity Server 100 provides the services of the failed server 102, while still protecting the remaining protected servers 102. …
>
> Integrity Server 100 is a conventional network computer node configured with a tape autoloader 110 (a tape "juke box" that automatically loads and unloads tape cartridges from a read/write head station), a disk 120, storage 130 (storage 130 is typically a portion of the disk, rather than RAM), and a programmed CPU (not shown).
>
> After a client node 104 updates a file of a file server 102, producing a new version of the file, the agent process on that file server 102 copies the new version of the file to the Integrity Server's disk 120. As the file is copied, a history package 140 is enqueued at the tail of an active queue 142 in the Integrity Server's storage 130; this history package 140 holds the data required for the Integrity Server's bookkeeping, for instance telling the original server name and file pathname of the file, its timestamp, and where the Integrity Server's current version of the file is stored. History package 140 will be retained in one form or another, and in one location or another (for instance, in active queue 142, offsite queue 160, or the catalog--see FIGS. 3a -3b) for as long as the file version itself is managed by Integrity Server 100.

When history package 140 reaches the head of active queue 142, the file version itself is copied from disk 120 to the current tape 150 in autoloader 110. History package 140 is dequeued to two places. History package 140 is enqueued to off-site queue 160 (discussed below), and is also stored as history package 312 in the protected files catalog, in a format that allows ready lookup given a "\\server\file" pathname, to translate that file pathname into a tape and an address on that tape at which to find the associated file version.

As tape 150 approaches full, control software unloads current tape 150 from the autoloader read/write station, and loads a blank tape as the new current tape 150. The last few current tapes 151-153 (including the tape 150 recently removed, now known as tape 151) remain in the autoloader as the "active set" so that, if one of servers 102 fails, the data on active set 150-153 can be accessed as stand-in copies of the files of the failed server 102.

When a file version is written to active tape 150, its corresponding history package 140 is dequeued from active queue 142 and enqueued in off-site queue 160. When an off-site history package 162 reaches the head of off-site queue 160, the associated version of the file is copied from disk 120 to the current off-site tape 164, and the associated history package 312 is updated to reflect the storage of the data to offsite media in the protected file catalog. History package 312 could now be deleted from disk 120. When current off-site tape 164 is full, it is replaced with another blank tape, and the previous off-site tape is removed from the autoloader, typically for archival storage in a secure off-site archive, for disaster recovery, or recovery of file versions older than those available on the legacy tapes.

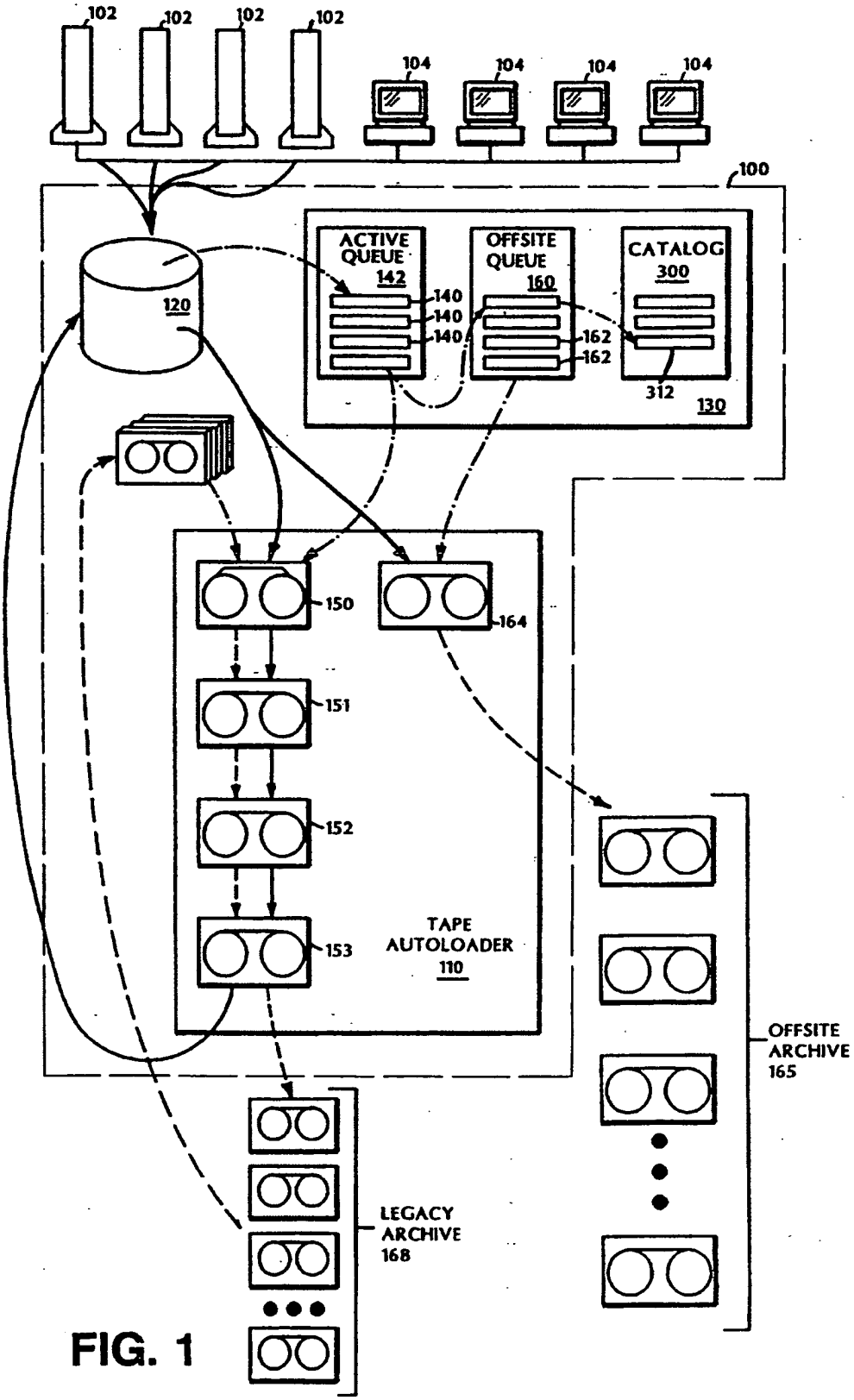
102
102
102
102
104
104
104
104
100
ACTIVE QUEUE 142
OFFSITE QUEUE 160
CATALOG 300
120
140
140
140
162
162
312
130
150
164
151
152
153
TAPE AUTOLOADER 110
OFFSITE ARCHIVE 165
LEGACY ARCHIVE 168

FIG. 1

3. Liu Overview

As generally taught by Liu in the Abstract:

An improved method of performing a high-performance backup of a computer system is described, which reduces disk read time and thus gains efficiency by reading input file blocks sequentially rather than the order in which the block appear in the original files. The improved method involves reading the working directory maintained by the operating system to determine all of the blocks associated with the set of files or other data aggregations to be backed up. The data block identities so determined are sorted in accordance with their physical location on the disk, thereby providing a sequential order for reading. The data to be backed up from the random access storage device or devices is read in this sequential order, and written to the backup media. There is also stored in conjunction with the backup media a Catalog containing the names of the files in the backup set, the location of the file data blocks on the backup media, the proper ordering of the blocks in this original file, and any other desired file attribute information.

4. Applicant's limitations recited in Claims 1, 9-13, and 21-24 are not taught, either individually or in combination, by Shannon, Midgely, or Liu

Claim 1 recites:

performing a full image backup in disk order on a plurality of data blocks stored by the at least one primary data source;

initiating an incremental backup at a predetermined interval, the incremental backup including file system metadata; and

comparing a modification time of each file/folder at the predetermined interval to a defined time, wherein if the modification time is earlier than the defined time, then excluding

data blocks of that file/folder from the incremental backup.

In contrast, Shannon teaches a computer file backup method rather than performing a full image backup. Although Shannon states that a "full image" can be created on tape or floppy disk (col. 1, lines 32-35), the actual backup method described by Shannon relates only to the saving and updating of files. The Office Action cites col. 3, lines 6-11 as teaching the image backup. Applicant traverses this characterization. Specifically, col. 3, lines 6-11 states:

> The method provides for creating a logical disk map of the client computer disk, and connecting the client computer to the server computer. The method further comprises copying a client computer logical disk image, including the logical disk map from the client computer disk, to the disk of the server computer.

This passage teaches nothing regarding performing a full image backup. In contrast, Applicant performs a full image backup in disk order on a plurality of data blocks. As noted by Applicant, because disk order (not file order) is used, an image backup can be significantly faster than a file-by-file backup. Paragraph [0004].

The Office Action further states that Shannon teaches file system metadata at col. 6, lines 38-44. Applicant traverses this characterization. Col. 6, lines 38-44 teaches:

> e) updating periodically by the client in said client computer said client disk map by comparing said client disk map with the previous client disk map to identify any client data file with additions, modifications or deletions occurring since the last update of the previous client disk map to provide a revised client disk map;

This passage teaches nothing regarding file system metadata. Specifically, to ensure an accurate incremental file update, Shannon further teaches creating a logical disk map of the client computer disk. Col. 3, lines 6-7. This logical disk map is copied to a disk of a server computer. Col. 3, lines 8-11. The logical disk map is periodically updated to create a list of modified/removed files. Col. 3, lines 17-23.

Using a map to facilitate the incremental backup adds significant system overhead. Specifically, as described as prior art by Applicant in paragraph [0005],

> In systems that want to provide image incremental backups, the additional software to track changes must be enabled. This software, at a minimum, must track which portion of the file system or storage has been re-written. This usually involves updating a map or a list tracking which blocks have been re-written. Thus, all write operations now require at least two writes: one write to update the change list or map and another write to write the data. Therefore, this method adds 100% overhead for writes on systems wanting to enable image incremental backups.

Note that a map update involves a map-to-map comparison, thereby adding considerable complexity and time to the update process. Applicant's technique advantageously eliminates the additional complexity and overhead of updating a map, like the logical disk map taught by Shannon.

Specifically, and recited in Claim 1, the modification time of each file/folder can be compared at the predetermined interval to a defined time. If the modification time is earlier than the defined time, then the data blocks of that file/folder can be excluded from the incremental backup.

As taught by Applicant in paragraph [0023], including file system metadata in the backup significantly increases the

accuracy of the backup compared to a standard file-by-file backup, which only identifies new/changed files. Moreover, because each file's/folder's modification time is already part of the file system metadata being tracked and updated by the file system, this backup method has no associated overhead during normal operation.

Notably, Shannon apparently distrusts this file system metadata. For example, Shannon teaches that the date on a vast majority of MS-DOS machines is an unreliable indicator because the date is poorly maintained. Col. 1, lines 63-65. Therefore, Shannon teaches away from Applicant's recited use of the modification time. In lieu of Applicant's simple but effective method, Shannon creates the elaborate and time-consuming logical disk map, as described above.

Midgely fails to remedy the deficiencies of Shannon. Specifically, Midgely also teaches a file-based backup. See, for example, col. 1, lines 59-61; col. 2, lines 34-36; and col. 2, lines 63-65. Therefore, Midegly teaches nothing regarding a full image backup in disk order. Midegly also teaches nothing regarding the use of file system metadata during an incremental backup.

The Office Action states that Midgely at col. 2, lines 17-19 and 37-39 teaches comparing the modification time of each file/folder at the predetermined interval to a defined time, wherein if the modification time is earlier than the defined time, then excluding data blocks of that file/folder from the incremental backup.

Col. 2, lines 17-19 teach:

> When file versions are dequeued, the queue is reviewed for later versions of the dequeued file: only the latest version of the dequeued file is actually written to the active volume, and other versions in the queue are purged.

Col. 2, lines 37-39 teach:

> In the method, recently-altered protected files are snapshotted to a storage cache. A new snapshot of a given file displaces any older snapshot of the same file from the storage cache.

Applicant submits that determining the later version of two files is different than comparing a modification time of each file/folder at a predetermined interval to a defined time. Therefore, Applicant traverses the characterization that Midgely teaches the step of comparing.

Liu also fails to remedy the deficiencies of Shannon and Midegely. Specifically, Liu also teaches a file-based backup. The Office Action states that Liu at col. 1, lines 62-64 and col. 5, lines 1-18 teaches an image backup in disk order. Applicant traverses this characterization. Col. 1, lines 62-64 teach:

> To achieve such reduction by performing sequential rather than random reads of the input file, to the extent feasible;

Col. 5, lines 1-18 teaches:

> The backup copying is performed as follows: Starting from the lowest sorted LCN number 445, the system reads the specified number of clusters (i.e., the corresponding Run-length), starting from the designated LCN, into a buffer, the size of which is preferably equal to an integral multiple of the cluster size 450. This read process (440) is repeated (advancing each time to the next LCN and Run), until a buffer-full of clusters has been read from the disk. A double or rotating buffer scheme is employed, and in order to completely fill buffers, Runs are split as necessary between the end of one buffer-full and the beginning of the next. Exact buffer size is a matter of tuning, and will vary from system to system.
>
> As each buffer-full is completed, the buffer is

written 455 to the backup medium, here assumed to be tape (though any storage medium could be used).

As buffers are written to tape, the Catalog information is updated 460 with the corresponding tape Block and Offset numbers, so that files and their data components can be located and restored from the backup set.

These passages, which are consistent with other passages not cited in the Office Action (e.g. col. 1, lines 58-67, col. 2, lines 1-5, col. 3, lines 36-40, and col. 5, lines 54-58), indicate that Liu teaches a file-based backup. Therefore, like Shannon and Midgely, Liu also teaches nothing regarding an image backup in disk order.

Because none of the cited references teach an image backup in disk order, Applicant submits that Claim 1 is patentable over such cited references.

Claims 9-12 depend from Claim 1 and therefore are patentable for at least the reasons presented for Claim 1.

Moreover, Claim 10 recites, "wherein the full backup and the incremental backup are used to provide a point-in-time disaster recovery." Claim 11 recites, "wherein the full image backup and the incremental backup are used to keep a standby machine up-to-date as of a last backup." Claim 12 recites, "wherein the full image backup and the incremental backup are written directly over a network to a standby machine and recovered, thereby keeping the standby machine up-to-date as of a last backup."

Because none of the cited references teach a full image backup in disk order (Applicant traverses any characterization in the Office Action to the contrary), it logically follows that these cited references also cannot teach further limitations of

that full image backup. Therefore, Claims 10-12 are further patentable based on their recited limitations.

Claim 13 recites,

> performing a full image backup in disk order on a plurality of data blocks stored by the at least one primary data source;
> initiating an incremental backup at a predetermined interval, the incremental backup including file system metadata; and
> comparing a modification time of each file/folder at the predetermined interval to a defined time, wherein if the modification time is later than the defined time, then including data blocks of that file/folder in the incremental backup.

Therefore, Claim 13 is patentable for substantially the same reasons presented for Claim 1.

Claims 21-23 depend from Claim 13 and therefore are patentable for at least the reasons presented for Claim 13.

Moreover, Claim 22 recites, "wherein the full backup and the incremental backup are used to provide a point-in-time disaster recovery." Claim 23 recites, "wherein the full image backup and the incremental backup are used to keep a standby machine up-to-date as of a last backup."

Because none of the cited references teach a full image backup in disk order (Applicant traverses any characterization in the Office Action to the contrary), it logically follows that these cited references also cannot teach further limitations of that full image backup. Therefore, Claims 22-23 are further patentable based on their recited limitations.

5. Distinguishing Between Full Image Backup And Backing Up Of Files.

The Office Action states that a "full image backup" is not distinguishable from a "file by file backup". Specifically, the Office Action states:

> [I]f only certain files are specified for backup, then a file by file backup reads on a "full image backup ... on a plurality of data blocks". In other words, the original claim language did not adequately address this distinction.

Applicant respectfully traverses this argument. As taught by Applicant in paragraphs [0003] and [0004] of the Specification,

> [0003] Generally, conventional backup methods provide for either file-by-file backup or image backup. In a file-by-file backup, the backup program copies one file at a time from the disk to the tape. Specifically, the program places all pieces of data for each file, irrespective of actual locations on the disk, into a single sequential block that is stored on the tape. Thus, a file-by-file backup can easily provide an incremental backup, wherein only those files that have been modified or added since the last backup are written to tape. However, a file-by-file backup fails to ensure that all changes to the files are noted. Specifically, the file-by-file backup fails to indicate removes (wherein a file has actually been deleted), renames (wherein the file is renamed), or links (wherein a file, such as an email, includes pointers to other files, e.g. other mail boxes). It also can be slow since files are written to tape in file order not disk order.
>
> [0004] In an image backup, the data image is read sequentially from the disk and written to the tape. Because disk order (not file order) is used, an image backup can be significantly faster than a file-by-file backup. Image backups have most often been used for full backups only. Image incremental backups exist today but are based on block-change lists. That is, an additional software layer must be used at the file system layer or at the device driver layer that tracks changes to underlying storage on a

> per block basis. Typically, when a portion of a file is re-written, the data can be written directly over the old data.

This distinction is also known and accepted in the industry. For example, Liu (U.S. Patent 6,415,300) teaches in col. 1, lines 14-30:

> Prior art backup methods generally provide for an "image" backup of an entire disk volume, or a "file-by-file" backup. An image backup copies the entire disk volume without regard to directory structure, and can be performed relatively quickly, although it does require time and space to copy the entire disk. However, since an image backup generally does not take account of directory and file information, such a backup does not support selective restoration of files. In order to be able to restore files selectively, generally a file-by-file backup has been required.
>
> Conventionally, the files to be backed up in a file-by-file backup are accessed in accordance in the normal manner provided by the operating system, in which data is read from the disk in the logical order of file contents. The actual physical blocks of data on the disk corresponding to each file are not, however, generally stored in a contiguous or linear order.

Because this terminology is supported by the Specification and clearly understood/accepted in the industry, Applicant submits that a "full image backup" is adequately distinguished from a "file-by-file backup".

6. Claims 1 and 13 Do Recite Disk Order

The Office Action indicates that Claims 1 and 13 do not recite writing data blocks in disk order (thereby increasing speed and accuracy compared to writing data blocks in file order). Applicant disagrees. Claims 1 and 13, as amended,

recite "performing a full image backup in disk order on a plurality of data blocks". Therefore, Applicant submits that there is no divergence between the Specification, paragraph [0023] and Claims 1 and 13.

(B) Claims 2-4 and 14-16 are patentable under 35 U.S.C. 103(a) over Shannon, Midgely, and Liu in view of the publication entitled "Oracle 7 Server Administrator's Guide" (Oracle).

1. Shannon, Midgely, Liu Overviews (see above)

2. Oracle Overview

Oracle teaches how to back up data in an ORACLE database. Page 18-1. To perform a full backup, Oracle teaches that all files used by the database must be closed by shutting down the database. Page 18-8. After this step is performed, the operating system commands or a backup utility can make backups of all data files, online redo log files, and a single control file of the database. Page 18-8. Once all data files are backed up, the database can be restarted. Page 18-8.

3. Applicant's limitations recited in Claims 2-4 and 14-16 are not taught, either individually or in combination, by Shannon, Midgely, Liu, or Oracle

Oracle fails to remedy the deficiencies of Shannon, Midgely, and Liu. Specifically, Oracle also teaches backing up files in the database. Page 18-8. Therefore, Oracle teaches nothing regarding a full image backup in disk order. Applicant submits that the "complete export" taught by Oracle (see, page 18-18) refers to exporting data files contained in the database, not the full image backup recited in the claims.

Claims 2-4 depend from Claim 1 and therefore are patentable for at least the reasons presented for Claim 1. Moreover, Claim 2 recites, "wherein the defined time is a time when the full image backup was performed." Claim 4 recites, "wherein the defined time is one of a first time when the full image backup was performed and a second time when a last incremental backup was performed, whichever is the more recent."

Because none of the cited references teach a full image backup in disk order (Applicant traverses any characterization in the Office Action to the contrary), it logically follows that these cited references also cannot teach further limitations of that full image backup. Therefore, Claims 2 and 4 are further patentable based on their recited limitations.

Claims 14-16 depend from Claim 13 and therefore are patentable for at least the reasons presented for Claim 13. Moreover, Claim 14 recites, "wherein the defined time is a time when the full image backup was performed." Claim 16 recites, "wherein the defined time is one of a first time when the full image backup was performed and a second time when a last incremental backup was performed, whichever is the more recent."

Because none of the cited references teach a full image backup in disk order (Applicant traverses any characterization in the Office Action to the contrary), it logically follows that these cited references also cannot teach further limitations of that full image backup. Therefore, Claims 14 and 16 are further patentable based on their recited limitations.

(C) Claims 5 and 17 are patentable under 35 U.S.C. 103(a) as being unpatentable over Shannon, Midegly, and Liu in view of U.S. Patent 5,195,025 (Boecker).

1. Shannon, Midgely, Liu Overviews (see above)

2. Boecker Overview

As taught by Boecker in col. 1, line 64 to col. 2, lines 21:

> [A] system and method ... interact with operating system and subsystem timer service(s) to gradually adjust software time-of-day clocks to comprehend seasonal time changes such as the one hour time change from Daylight Savings to Standard Time and again back to Daylight Savings Time. A preferred embodiment changes the operating system's interpretation of the time-of-day clock by gradually modifying the time services control blocks, also known as offsets, such that application programs executing in the system will always perceive ascending time-of-day values to internal requests for time of day.
>
> Very briefly, the preferred embodiment of the present invention transfers control to a first module. This first module performs initialization and then invokes a first submodule to perform the necessary IMS initialization function. The first module then enters a loop where the gradual time change occurs. Each iteration of the loop results in the modification of the MVS time-related control blocks and an invocation of a second submodule for the modification of the IMS time-related storage. When the first module determines that the time has been changed by the required amount, it performs its termination function and invokes a third submodule to perform the necessary IMS termination function.

3. Applicant's limitations recited in Claims 5 and 17 are not taught, either individually or in combination, by Shannon, Midgely, Liu, or Boecker.

Boecker fails to remedy the deficiencies of Shannon, Midgely, and Liu. Specifically, Boecker teaches nothing regarding a full image backup in disk order. Claim 5 depends from Claim 1 and therefore is patentable for at least the reasons presented for Claim 1. Moreover, Claim 5 recites, "determining whether a system clock has been changed".

The Office Action states that the Abstract of Boecker teaches this limitation. Applicant traverses this characterization. As Boecker teaches in the Abstract:

> A system and method for dynamically changing a computer's time-related control blocks to coincide with seasonal time-of-day changes is shown including a first module for synchronizing the time-related control blocks and a second module connected to said first module for monitoring transfer of control of the time-related conrol blocks between the computer and the first module. Another system and method for dynamically changing a central processing unit time-of-day clock, without interrupting applications executing concurrently and without incurring any system down time entails transferring control of a time-of-day interpreter employing an offset to a time changer module, modifying the offset by a predetermined value at a predetermined rate until the offset reaches its synchronization value, and returning control of the interpreter to the computer. Other devices, systems and methods are also disclosed.

The Abstract of Boecker fails to teach determining whether a system clock has been changed. Therefore, Claims 5 and 17 are further patentable based on their recited limitations.

(D) Claims 6-8 and 18-20 are patentable under 35 U.S.C. 103(a) as being unpatentable over Shannon, Midegly, Liu, and Boecker in view of Oracle.

1. Shannon, Midgely, Liu, Boecker, Oracle Overviews (see above)

2. Applicant's limitations recited in Claims 6-8 and 18-20 are not taught, either individually or in combination, by Shannon, Midgely, Liu, Boecker, or Oracle.

Claims 6 and 18 depend from Claims 5 and 17, respectively, and therefore are patentable for at least the reasons presented for Claims 5 and 17. Moreover, Claims 6 and 18 recite, "wherein if the system clock has been changed, then returning to performing the full image backup on the plurality of data blocks". The Office Action states that Oracle at pages 18-3, 18-18, and 18-19 teach this limitation. Applicant traverses this characterization. As noted above, Oracle teaches a file-based backup and therefore cannot teach returning to performing the full image backup, as recited in Claims 6 and 18. Therefore, Claims 6 and 18 are further patentable based on their recited limitations.

Claims 7 and 19 depend from Claims 6 and 18, respectively, and therefore are patentable for at least the reasons presented for Claims 6 and 18. Moreover, Claims 7 and 19 recite, "wherein if the system clock has not been changed, then initiating the incremental backup at the predetermined interval". The Office Action states that Oracle at pages 18-3, 18-18, and 18-19 teach this limitation. Applicant traverses this characterization. Oracle teaches nothing about initiating in incremental backup at a predetermined interval if a system clock has not been changed, as recited in Claims 7 and 19. Therefore, Claims 7 and 19 are further patentable based on their recited limitations.

Claims 8 and 20 depend from Claims 6 and 18, respectively, and therefore are patentable for at least the reasons presented for Claims 6 and 18. Moreover, Claims 8 and 20 recite, "wherein if the system clock has not been changed, then comparing the modification time of each file/folder at the predetermined interval to the defined time". The Office Action states that Oracle at pages 18-3, 18-18, and 18-19 teach this limitation. Applicant traverses this characterization. Oracle teaches nothing about comparing the modification time of each

file/folder at the predetermined interval to the defined time if a system clock has not been changed, as recited in Claims 8 and 20. Therefore, Claims 8 and 20 are further patentable based on their recited limitations.

IX. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejections of Claims 1-24 are erroneous, and reversal of these rejections is respectfully requested.

Respectfully submitted,

Customer No.: 22888

Jeanette S. Harms
Attorney for Applicant
Reg. No. 35,537

I hereby certify that this correspondence is being deposited with the United States Postal Service as FIRST CLASS MAIL in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on March 28, 2005.

3/28/2005
Date

Rebecca A. Baumann
Signature Rebecca A. Baumann

X. APPENDIX A

1. (Previously Amended) A method for backing up data in a computer system from at least one primary data source to a secondary data source, the method comprising:

performing a full image backup in disk order on a plurality of data blocks stored by the at least one primary data source;

initiating an incremental backup at a predetermined interval, the incremental backup including file system metadata; and

comparing a modification time of each file/folder at the predetermined interval to a defined time, wherein if the modification time is earlier than the defined time, then excluding data blocks of that file/folder from the incremental backup.

2. (Original) The method of Claim 1, wherein the defined time is a time when the full image backup was performed.

3. (Original) The method of Claim 1, wherein the defined time is a time when a last incremental backup was performed.

4. (Original) The method of Claim 1, wherein the defined time is one of a first time when the full image backup was performed and a second time when a last incremental backup was performed, whichever is the more recent.

5. (Original) The method of Claim 1, further including determining whether a system clock has been changed.

6. (Original) The method of Claim 5, wherein if the system clock has been changed, then returning to performing the full image backup on the plurality of data blocks.

7. (Original) The method of Claim 6, wherein if the system clock has not been changed, then initiating the incremental backup at the predetermined interval.

8. (Original) The method of Claim 6, wherein if the system clock has not been changed, then comparing the modification time of each file/folder at the predetermined interval to the defined time.

9. (Previously Amended) The method of Claim 1, wherein the file system metadata allows the tracking of new, changed, renamed, and linked files/folders.

10. (Original) The method of Claim 1, wherein the full backup and the incremental backup are used to provide a point-in-time disaster recovery.

11. (Original) The method of Claim 1, wherein the full image backup and the incremental backup are used to keep a standby machine up-to-date as of a last backup.

12. (Original) The method of Claim 1, wherein the full image backup and the incremental backup are written directly over a network to a standby machine and recovered, thereby keeping the standby machine up-to-date as of a last backup.

13. (Previously Amended) A method for backing up data in a computer system from at least one primary data source to a secondary data source, the method comprising:

performing a full image backup in disk order on a plurality of data blocks stored by the at least one primary data source;

initiating an incremental backup at a predetermined interval, the incremental backup including file system metadata; and

comparing a modification time of each file/folder at the predetermined interval to a defined time, wherein if the modification time is later than the defined time, then including data blocks of that file/folder in the incremental backup.

14. (Original) The method of Claim 13, wherein the defined time is a time when the full image backup was performed.

15. (Original) The method of Claim 13, wherein the defined time is a time when a last incremental backup was performed.

16. (Original) The method of Claim 13, wherein the defined time is one of a first time when the full image backup was performed and a second time when a last incremental backup was performed, whichever is the more recent.

17. (Original) The method of Claim 13, further including determining whether a system clock has been changed.

18. (Original) The method of Claim 17, wherein if the system clock has been changed, then returning to performing the full image backup on the plurality of data blocks.

19. (Original) The method of Claim 18, wherein if the system clock has not been changed, then initiating the incremental backup at the predetermined interval.

20. (Original) The method of Claim 18, wherein if the system clock has not been changed, then comparing the modification time of each file/folder at the predetermined interval to the defined time.

21. (Previously Amended) The method of Claim 13, wherein the file system metadata allows the tracking of new, changed, renamed, and linked files/folders.

22. (Original) The method of Claim 13, wherein the full backup and the incremental backup are used to provide a point-in-time disaster recovery.

23. (Original) The method of Claim 13, wherein the full image backup and the incremental backup are used to keep a standby machine up-to-date as of a last backup.

24. (Original) The method of Claim 13, wherein the full image backup and the incremental backup are written directly over a network to a standby machine and recovered, thereby keeping the standby machine up-to-date as of a last backup.

PTO/SB/21 (09-04)
Approved for use through 07/31/2006. OMB 0651-0031
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/066,109
Filing Date	01/31/2002
First Named Inventor	Daniel D. McNeil
Art Unit	2171
Examiner Name	Patrick JD Santos
Total Number of Pages in This Submission: 34	**Attorney Docket Number** MPT-006

ENCLOSURES *(check all that apply)*

- ☒ Fee Transmittal Form
  - ☐ Fee Attached
- ☐ Amendment / Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Reply to Missing Parts/Incomplete Application
  - ☐ Reply to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Terminal Disclaimer
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Statement Under 37 CFR 3.73(b)
- ☐ CD, Number of CD(s) _____
- ☐ Request for Refund

Remarks

- ☐ After Allowance Communication to TC
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☒ Appeal Communication to TC *(Appeal Notice, Brief, Reply Brief)*
- ☐ Proprietary Information
- ☐ Status Letter
- ☒ Other Enclosure(s) *(please identify below):*

Return Receipt Postcard

SIGNATURE OF APPLICANT, ATTORNEY OR AGENT

Firm Name	BEVER, HOFFMAN & HARMS, LLP	Customer Number	35,537
Signature			
Printed Name	Jeanette S. Harms		
Date	March 28, 2005	Reg. No.	35,537

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-PatentsCommissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on March 28, 2005.

Signature			
Typed or printed name	Rebecca A. Baumann	Date	March 28, 2005

PTO/SB/17 (12-04)
Approved for use through 07/31/2006. OMV 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Effective on 12/08/2004
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818)

FEE TRANSMITTAL For FY 2005

☒ Applicant claims small entity status. See 37 C.F.R. § 1.27

TOTAL AMOUNT OF PAYMENT ($) 250.00

Complete if Known	
Application Number	10/066,109
Filing Date	01/31/2002
First Named Inventor	Daniel D. McNeil
Examiner Name	Patrick JD Santos
Art Unit	2171
Attorney Docket No	MPT-006

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit Card ☐ Money Order ☐ None ☐ Other (please identify):_____

☒ Deposit Account Deposit Account Number: 50-0574 Deposit Account Name: **Bever, Hoffman & Harms, LLP**

For the above-identified deposit account, the Director is hereby authorized to; (check all that apply)

☒ Charge fee(s) indicated below ☐ Charge fee(s) indicated below, except for the filing fee

☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 ☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	**Fees Paid ($)**
Utility	300	150	500	250	200	100	$
Design	200	100	100	50	130	65	$
Plant	200	100	300	150	160	80	$
Reissue	300	150	500	250	600	300	$
Provisional	200	100	0	0	0	100	$

2. EXCESS CLAIM FEES

Fee Description	Fee($)	Small Entity Fee($)
Each claim over 20 or, for Reissues, each claim over 20 and more than in the original patent	50	25
Each independent claim over 3 or, for Reissues, each independent claim more than in the original patent	200	100
Multiple dependent claims	360	180

Total Claims	Extra Claims	Fee($)	**Fee Paid ($)**	Multiple Dependent Claims	
	- 20 or HP =	x	=	Fee($)	Fee($)

HP = highest number of total claims paid for, if great than 20

Indep. Claims	Extra Claims	Fee($)	**Fee Paid ($)**
	- 3 or HP =	x	=

HP = highest number of total claims paid for, if great than 3

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 USC 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee($)	Fee Paid ($)
- 100 =	5- = (round up to a whole number) x		=	

4. OTHER FEE(S)

	Fee Paid ($)
Non-English Specification - $130 fee (no small entity discount)	
Other: Appeal Brief $250.00	

SUBMITTED BY

Signature:	Registration No. 35,537	Telephone:	(408) 451-5907
Name (Print/Type) Jeanette S. Harms		Date:	March 28, 2005